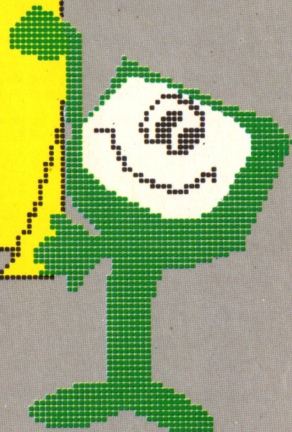
VIDEO BASIC

20 VIDEOLEZIONI DI BASIC
PER IMPARARE CON L'MSX

- TV e monitor
- Schermo e memoria video
- COLOR
- Funzioni di controllo della stampa su video
- SCREEN, WIDTH, SPC,
- TAB, POS, CSRLIN, LOCATE
- Numeri a caso con RND
- L'arte di stampare
- Videoesercizi
- Videogioco n. 6

VIDEOBASIC MSX
Pubblicazione quattordicinale
edita dal Gruppo Editoriale Jackson
Direttore Responsabile:
Giampietro Zanga
Direttore e Coordinatore
Editoriale: Roberto Pancaldi
Autore: Softidea -
Via Indipendenza 88-90 - Como
Redazione software:
Michele Casartelli
Francesco Franceschini
Progetto grafico:
Studio Nuovidea - via Longhi, 16 - Milano
Impaginazione:
Moreno Confalone
Illustrazioni:
Cinzia Ferrari, Silvano Scolari
Fotografie:
Marcello Longhini
Distribuzione: SODIP
Via Zuretti, 12 - Milano
Fotocomposizione: Lineacomp S.r.l.
Via Rosellini, 12 - Milano
Stampa: Grafika '78
Via Trieste, 20 - Pioltello (MI)
Direzione e Redazione:
Via Rosellini, 12 - 20124 Milano
Tel. 02/6880951/5

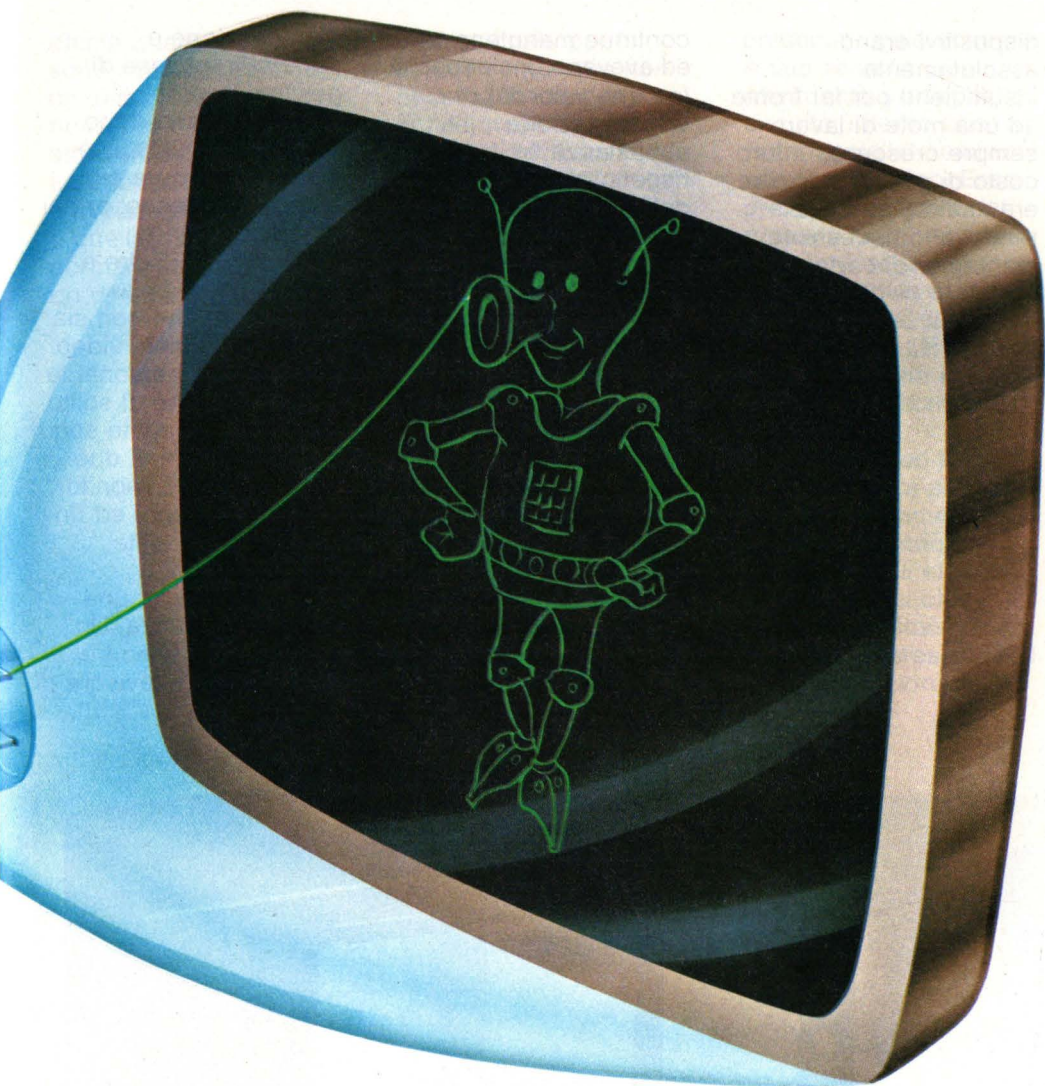
Tutti i diritti di riproduzione e pubblicazione di disegni, fotografie, testi sono riservati.
© Gruppo Editoriale Jackson 1985.
Autorizzazione alla pubblicazione Tribunale di Milano n° 422 del 22-9-1984
Spedizione in abbonamento postale Gruppo II/70 (autorizzazione della Direzione Provinciale delle PPTT di Milano).
Prezzo del fascicolo L. 8.000
Abbonamento comprensivo di 5 raccoglitori L. 165.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12 20124 Milano, mediante emissione di assegno bancario o cartolina vaglia oppure utilizzando il c.c.p. n° 11666203.
I numeri arretrati possono essere richiesti direttamente all'editore inviando L. 10.000 cdu. mediante assegno bancario o vaglia postale o francobolli.
Non vengono effettuate spedizioni contrassegno.

SOMMARIO

Introduzione

Il tuo computer è, tra le altre cose, anche una speciale stazione emittente televisiva. È infatti in grado di trasmettere via cavo le informazioni elaborate. Sotto questo aspetto, il programmatore diventa il "regista" dell'output dei dati trattati.
A lui sono, cioè, demandate le responsabilità di dare alle informazioni in uscita il massimo risalto e la maggiore chiarezza possibile.
Da qui l'importanza di conoscere sia l'hardware dedicato alla visualizzazione delle immagini, che le istruzioni e funzioni del BASIC capaci di controllare il formato e gli attributi.
Molto del successo dei tuoi programmi futuri dipende dalla familiarità con questi elementi.

Televisori e monitor

Il televisore ed il monitor - o, più generalmente, le unità video - costituiscono il principale dispositivo di uscita di un computer. Ad esse, infatti, viene normalmente affidato il compito di visualizzare tutte le informazioni, i dati ed i messaggi che costituiscono la base del fondamentale rapporto di interazione tra l'uomo e l'elaboratore.
Le funzioni svolte da un'unità video sono fondamentalmente tre:
1) visualizzare sullo schermo la maggior parte dei caratteri battuti sulla tastiera (tale funzione è chiamata eco);
2) visualizzare l'output dei programmi, consentendo di risparmiare tempo e carta quando non è necessaria una copia permanente;
3) inviare all'utente i messaggi dell'interprete BASIC (ad esempio le segnalazioni di errore).
L'uso delle unità video come dispositivi di output è abbastanza recente: fino a pochi anni fa le informazioni in uscita dal calcolatore erano infatti visualizzate quasi esclusivamente da stampanti e telescriventi. Ben presto, però, ci si rese conto che tali

HARDWARE

dispositivi erano assolutamente insufficienti per far fronte ad una mole di lavoro sempre crescente. Il loro costo di gestione, inoltre, era molto elevato (si rendevano necessarie montagne di carta e continue manutenzioni) ed avevano una capacità (ed una velocità) di visualizzazione abbastanza limitata rispetto alle esigenze dell'utenza.

Fu così che si pensò di affiancare alle pur necessarie stampanti delle unità di output più adeguate e flessibili all'uso di quanto non fossero quelle utilizzate fino a quel momento.

La scelta, come certamente avrai già immaginato, cadde sugli schermi video.

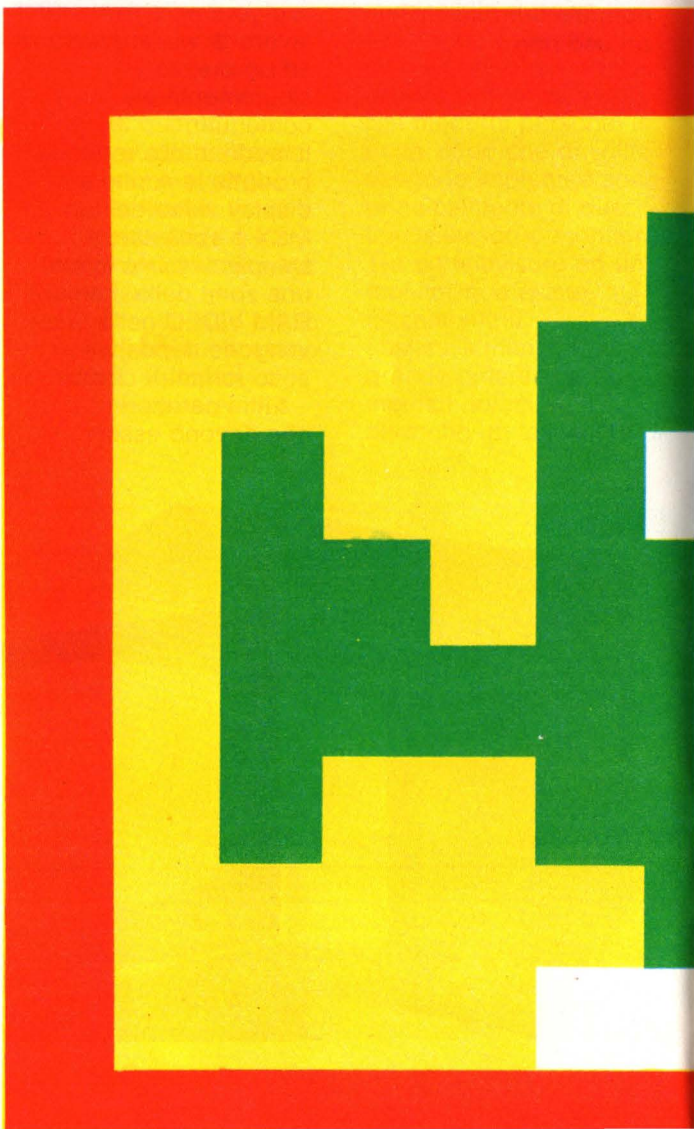
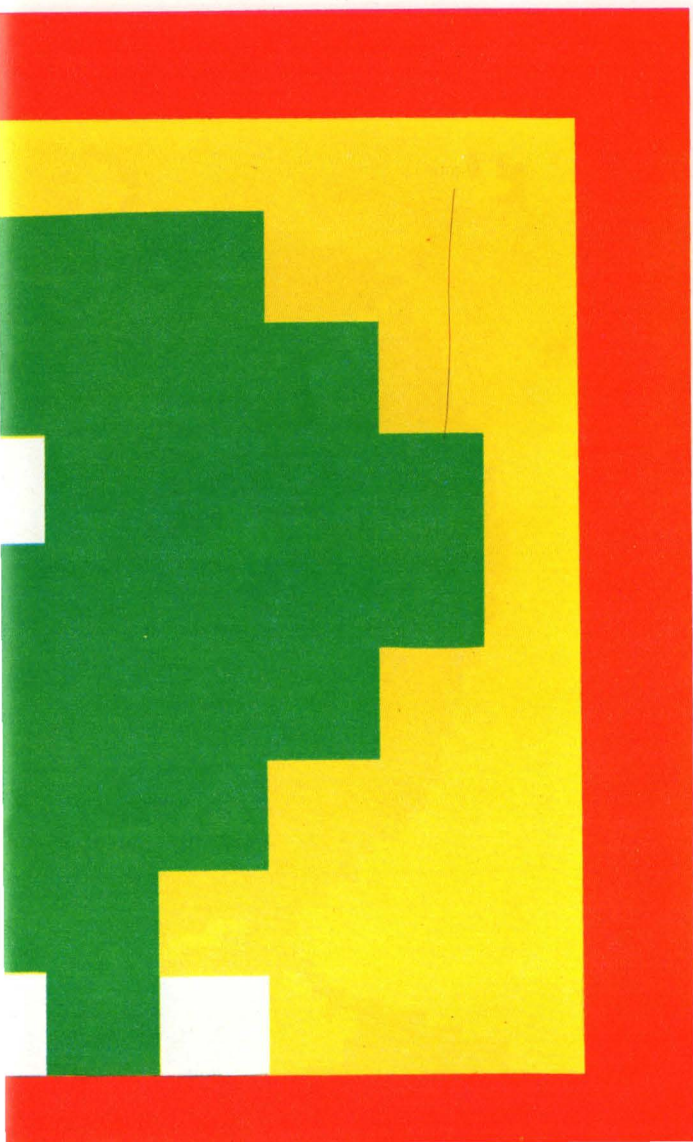
Essi rispondevano a tutti i requisiti richiesti: erano compatti, affidabili, economici (scarsa manutenzione e ridottissime spese di gestione), veloci.

Da allora, anno dopo anno, il loro utilizzo è diventato sempre più esteso ed intenso: ormai ai giorni nostri è impossibile riuscire a trovare un qualsiasi calcolatore che non sia provvisto di unità video.

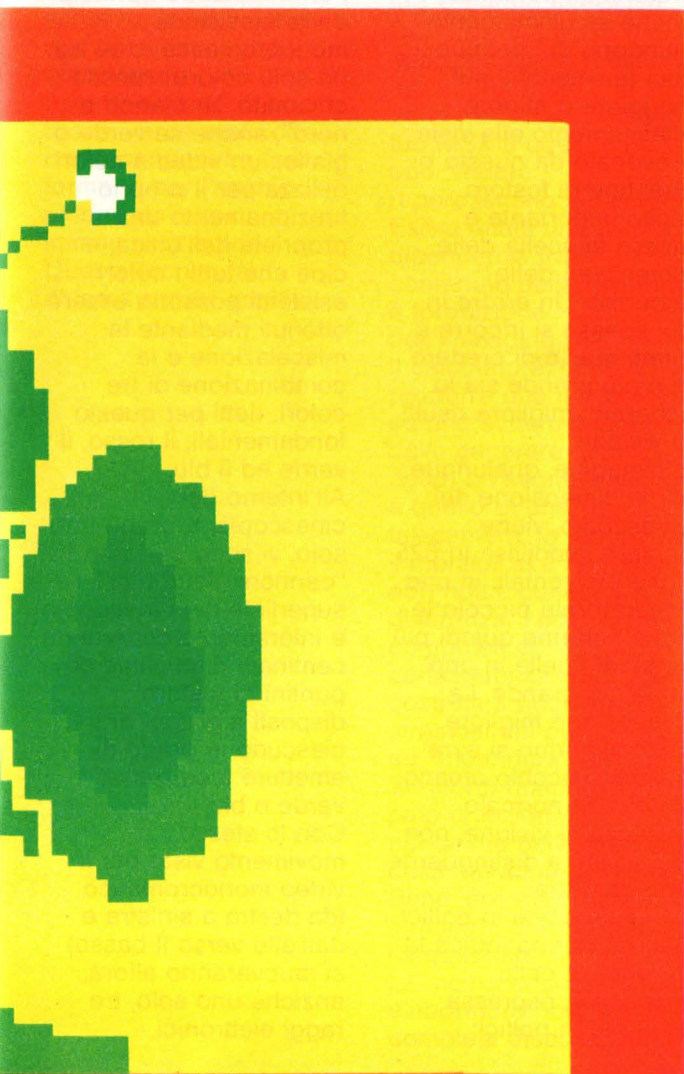
Nei moderni personal le unità video che di solito vengono impiegate sono principalmente di due tipi: televisori e monitor.

Tra un televisore ed un monitor non esiste, fisicamente e sostanzialmente, una grande differenza. Un monitor, infatti, non è altro che un televisore di

ottima qualità al quale sono stati asportati tutti i circuiti adatti per la ricezione dei segnali attraverso l'antenna.
La qualità dell'immagine è ovviamente superiore a quella ottenibile da un normale televisore. Per un uso non professionale può essere comunque non necessario (se non inutile) ricorrere all'acquisto di un monitor: anche il televisore di casa è infatti in grado di svolgere egregiamente il lavoro di visualizzazione, ad un prezzo sicuramente più contenuto.
Il modo in cui vengono prodotte le scritte sul display video del tuo MSX è abbastanza semplice: esiste infatti una zona della memoria RAM VIDEO nella quale vengono depositati - sotto forma di codice - tutti i caratteri che devono essere presentati sullo schermo.
A tale zona fa riferimento un apposito circuito, interno che "interfaccia" il computer con l'unità video. Esso preleva cioè tutti i dati presenti nelle varie locazioni della memoria video e li invia al circuito del televisore, sotto forma di impulsi elettrici compatibili con il sistema o lo standard televisivo.
Il televisore (od il monitor video) produce quindi una immagine visibile partendo da questi segnali elettrici.
Il modo in cui ciò avviene costituisce una delle più interessanti ed utili applicazioni della

fisica elettronica e vede come componente principale un dispositivo chiamato cinescopio.
Un cinescopio (o'tubo catodico) è un tubo a vuoto, cioè un contenitore di vetro nel quale è stato fatto il vuoto.
All'interno del tubo si trova un "cannone elettronico" (basato su un filamento riscaldato dalla corrente che lo attraversa, come nelle lampadine) in grado di produrre un fascio (o pennello) di elettroni molto sottile.
Gli elettroni godono di una singolare proprietà:

quando colpiscono particolari sostanze fluorescenti - chiamate anche fosfori - fanno generare a queste sostanze una luminescenza, la cui durata può andare da alcuni millisecondi (millesimi di secondo) ad alcuni secondi, in funzione del tipo di fosforo e dell'intensità del raggio elettronico. L'immagine viene ricostruita proprio sfruttando questo fatto: il fascio di elettroni, "sparato" dal cannone e comandato da opportuni campi elettrici e magnetici, spostandosi da destra a sinistra e dall'alto verso il basso applica maggiore o minore intensità ai singoli punti di uno schermo che è stato ricoperto da un sottile strato di fosfori, provocandone una maggiore o minore luminosità.

**La bassa risoluzione utilizza esclusivamente i caratteri (normali o grafici) presenti sulla tastiera.
Le immagini così ottenute non risultato molto dettagliate.**

L'immagine viene pertanto ricostruita punto per punto dal pennello elettronico del tubo catodico alla stessa maniera in cui l'occhio di un uomo legge la pagina stampata di un giornale o di un libro. La velocità con cui tale pennello attraversa l'intero schermo è però talmente elevata da non poter essere minimamente percepita dall'osservatore umano: lo standard televisivo europeo prevede infatti che l'intero schermo, suddiviso in 625 righe orizzontali (mentre nello standard americano le

linee sono 525) sia percorso completamente ogni cinquantesimo di secondo. Il fenomeno della persistenza dell'immagine sulla retina fornisce quindi l'illusione di una immagine completa e simultanea.

All'interno del televisore (o del monitor) vi sono, come accennato, anche dei circuiti destinati a comandare i movimenti del raggio elettronico sia in senso orizzontale che verticale.
Perché l'immagine sia visibile è però necessario che questi dispositivi lavorino simultaneamente.
Nel segnale video sono allora compresi, oltre alle informazioni riguardo all'intensità di ogni singolo punto dello schermo, anche appositi segnali destinati a coordinare il movimento del pennello elettronico. Sono i segnali di sincronismo.
Nel caso della trasmissione televisiva tutte queste informazioni provengono da una telecamera; per i computer, invece, esiste un apposito circuito che, proprio come una telecamera, "legge"

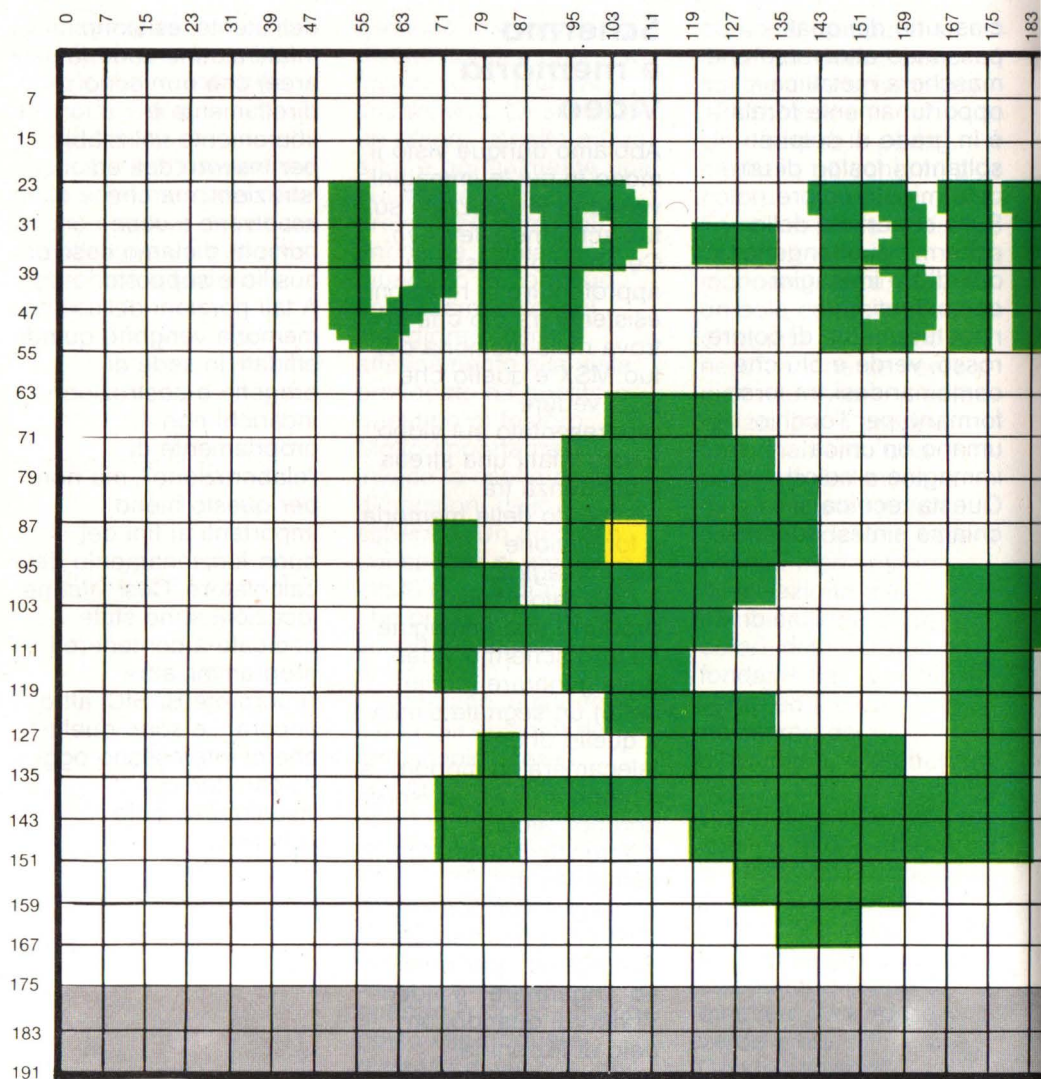
Con l'alta risoluzione puoi indirizzare il singolo pixel determinandone o meno l'accensione.
Le immagini così ottenute risultano molto dettagliate.

l'immagine da visualizzare nelle varie locazioni riservate alla memoria video.
Il tipo di fosforo applicato sulla superficie dello schermo determina anche il colore del punto di collisione tra pennello elettronico e fosforo stesso.

Esistono in commercio diversi tipi di monitor: a fosfori verdi, bianchi, ambra, ecc. La scelta di un colore rispetto ad un altro è normalmente una questione di gusti e preferenze personali, anche se ultimamente giungono di continuo voci (e smentite) sul maggiore o minore affaticamento alla vista provocato da questo o quel tipo di fosforo.
Molto importante è invece la scelta delle dimensioni dello schermo. Un errore in cui spesso si incorre è infatti quello di credere che più grande sia lo schermo, migliore risulti la visione.
L'immagine, qualunque sia la dimensione del cinescopio, viene sempre suddivisa in 625 righe orizzontali: in uno schermo più piccolo le righe saranno quindi più sottili di quelle in uno schermo grande. La dimensione migliore dello schermo si avrà quando l'occhio umano, posto alla normale distanza di visione, non riesce più a distinguerne una dall'altra.
La dimensione in pollici dello schermo indica la lunghezza della diagonale, espressa appunto in pollici: personal computer e terminali video hanno solitamente schermi da 12" (12 pollici).
Un discorso a parte meritano gli schermi a colori.
Fermi restando i principi della televisione monocromatica, cioè ad un solo colore (spesso chiamato "in bianco e nero", anche se verde o giallo), un video a colori utilizza per il proprio funzionamento una proprietà dell'ottica, e cioè che tutti i colori esistenti possono essere ottenuti mediante la miscelazione e la combinazione di tre colori, detti per questo fondamentali: il rosso, il verde ed il blu.
All'interno del cinescopio, anziché uno solo, vi sono pertanto tre "cannoni elettronici". La superficie del cinescopio è interamente coperta da centinaia di migliaia di puntini di fosforo disposti a gruppi di tre, ciascuno in grado di emettere luce rossa, verde o blu.
Con lo stesso movimento visto per il video monocromatico (da destra a sinistra e dall'alto verso il basso) si muoveranno allora, anziché uno solo, tre raggi elettronici,

ciascuno dei quali - passando attraverso una maschera metallica opportunamente forata - è in grado di colpire soltanto i fosfori di un determinato colore.
Sulla superficie dello schermo si ottengono quindi tre immagini coincidenti, rispettivamente di colore rosso, verde e blu che combinandosi tra loro formano per l'occhio umano un'unica immagine a colori.
Questa tecnica si chiama sintesi additiva.

Schermo e memoria video

Abbiamo dunque visto il modo in cui le immagini vengono visualizzate sul tuo schermo televisivo.
Cerchiamo adesso di approfondire la relazione esistente tra ciò che si trova nella memoria del tuo MSX e quello che puoi vedere rappresentato sul video.
Esiste infatti una stretta dipendenza tra contenuto della memoria e formazione dell'immagine.
Il calcolatore, per produrre una immagine su uno schermo video, deve generare (come detto) un segnale simile a quello di una telecamera, in modo che il monitor (o il televisore) non noti alcuna differenza. I costruttori di computer, di conseguenza, sono stati costretti a ricorrere a stratagemmi per riuscire ad "ingannare" il video.
Ti ricordi quando, un paio di lezioni fa, parlammo della mappa della rnemoria del MSX?
Bene, ora è arrivato il momento di rinfrescare l'argomento.
La memoria di un elaboratore non è a completa disposizione dell'utente: esistono infatti alcune zone (o aree) che non sono direttamente e liberamente utilizzabili per inserire dati ed istruzioni, ma che assolvono invece compiti, diciamo così, di ausilio e supporto.
A tali porzioni della memoria vengono quindi affidati, in sede di progetto e costruzione, incarichi non propriamente di "elaborazione", ma non per questo meno importanti ai fini del buon funzionamento del calcolatore. Così, alcune locazioni sono state dedicate a contenere i programmi, altre l'interprete BASIC, altre ancora - e sono quelle che ci interessano oggi - i caratteri da visualizzare sullo schermo.

L'area della memoria di cui ci vogliamo occupare prende il nome di memoria video. Il suo scopo è quello di contenere tutte le informazioni necessarie a costruire una

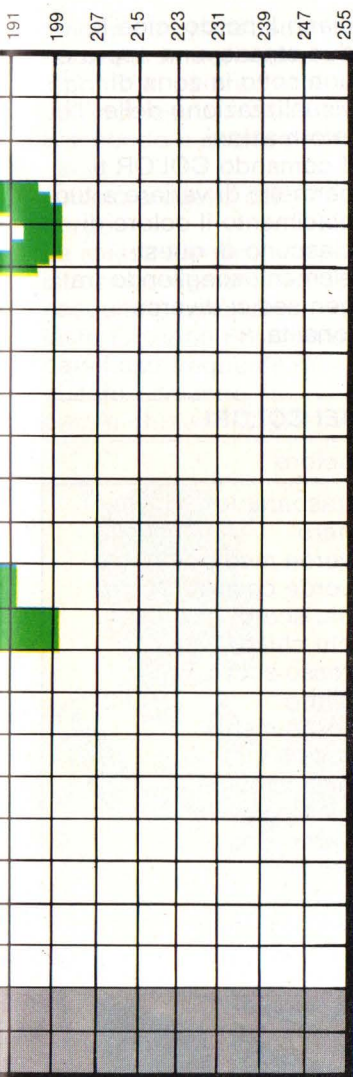



immagine sul display televisivo.
Secondo te, sarà una memoria RAM od una memoria ROM? Non dovresti avere molti dubbi.
Per forza di cose, infatti, dal momento che vogliamo poter modificare il contenuto dello schermo, la memoria video deve appartenere alla zona di memoria RAM. In essa devono quindi avvenire cambiamenti tutte le volte in cui, per esempio, esegui una istruzione PRINT o INPUT, oppure (a parte casi particolari) batti qualcosa sulla tastiera che deve essere visualizzato sullo schermo.

Da sola, però, la memoria video non può fare molto: è infatti necessario che un apposito circuito video, chiamato di refresh (cioè "rinfresco") dell'immagine, legga - esattamente come una telecamera - tutta la memoria, carattere per carattere.

A quel punto il circuito video, venuto a conoscenza dei codici dei caratteri da visualizzare, consulta una particolare zona della memoria dalla quale preleva la "descrizione" grafica del carattere stesso. Come risultato finale si ottiene quindi un segnale video indicante se ciascun punto dello schermo deve essere acceso o spento.
Tutto ciò che compare sul video viene infatti rappresentato sotto forma di particolari combinazioni di puntini luminosi, chiamati pixel (abbreviazione dell'inglese picture element).
Ciascun puntino, potendo essere solo acceso o spento, è pertanto descrivibile da un singolo bit.
Senza addentrarci ulteriormente in dettagli tecnici, ti basti sapere che il circuito di refresh "prende" la combinazione di bit contenente la descrizione del carattere e la mette nella corrispondente casella dello schermo, accendendo o spegnendo i puntini come indicato precedentemente.
In totale i pixel luminosi disponibili sul tuo MSX sono 49.152. Ad essi corrisponde nel complesso uno schermo composto da 192 minuscole righe di 256 pixel ciascuna.

COLOR

Quando accendi il tuo MSX il sistema operativo residente in ROM si preoccupa, tra le altre cose, di colorare in un modo particolare lo schermo del televisore: bianco l'inchiostro di scrittura dei caratteri; blu il colore della carta (sfondo); ciano il bordo, cioè le due strisce, una sopra e una sotto la zona di visualizzazione delle informazioni.
Il comando COLOR ti permette di variare a tuo piacimento il colore di ciascuno di questi elementi scegliendo tra ben sedici diverse tonalità.

TAVOLA DEI COLORI	
codice	colore
0	trasparente
1	nero
2	verde medio
3	verde chiaro
4	blu scuro
5	blu chiaro
6	rosso scuro
7	ciano
8	rosso medio
9	rosa
10	giallo scuro
11	giallo chiaro
12	verde scuro
13	magenta
14	grigio
15	bianco

All'accensione dunque, viene in pratica eseguita l'istruzione

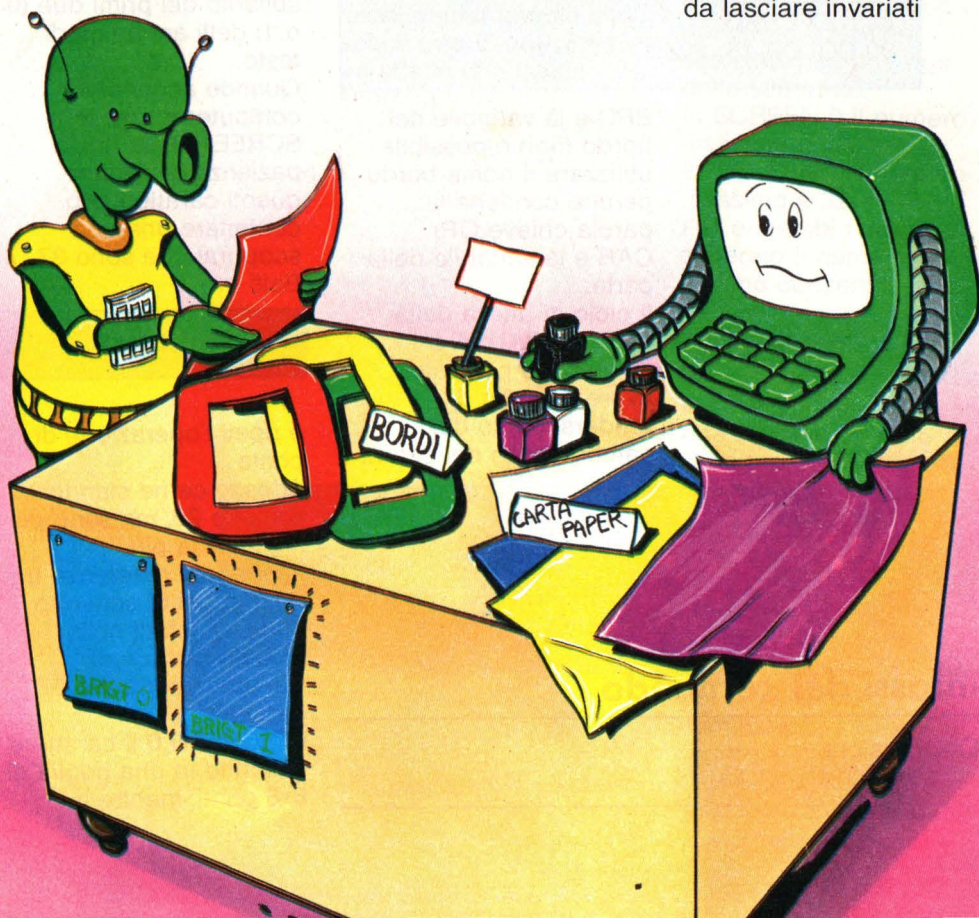
```
COLOR 15, 4, 7
```

poiché i tre parametri numerici si riferiscono:
il primo al colore dell'inchiostro;
il secondo a quello della carta;
l'ultimo a quello del bordo.
La stessa identica istruzione è subito eseguita premendo il tasto funzione F6; inoltre, dato l'uso frequente di questo comando, la parola chiave COLOR viene automaticamente stampata sullo schermo premendo F1.

```
COLOR 1, 15, 8
```

fissa il colore nero per l'inchiostro, bianco per la carta, rosso per il bordo.

```
COLOR 12, 12, 12
```

Lo schermo diventa tutto verde scuro. Questa situazione è analoga a quella ottenuta scrivendo con una matita verde su di un foglio verde: anche se il testo c'è, non si legge.
Particolare cura va applicata nella scelta dei colori dell'inchiostro e della carta per non affaticare inutilmente la vista.
È possibile modificare anche uno solo dei parametri del comando COLOR omettendo quelli da lasciare invariati

(attenzione però a non dimenticare le virgole).

```
COLOR ,,10
```

modifica soltanto il colore del bordo in giallo.

```
10 REM prova colori
20 FOR BRD=0TO15
30 COLOR,,BRD
40 FOR CAR = 0TO15
50 COLOR, CAR
60 FOR PAUSA = 0TO222 : NEXT PAUSA
70 NEXT CAR
80 NEXT BRD
90 COLOR 15, 1, 2
```

BRD è la variabile del bordo (non è possibile utilizzare il nome bordo perche contiene la parola chiave OR).
CAR è la variabile della carta.
Il ciclo di attesa della variabile PAUSA è necessario per poter avere il tempo di rendersi conto del cambiamento dei colori.

Sintassi del comando

COLOR valore
inchiostro, carta, bordo.

SCREEN

Il computer MSX dispone di 4 possibili configuarzioni dello schermo (modi).
In SCREEN 0 e in SCREEN 1 è possibile visualizzare soltanto caratteri, mentre per poter accedere all'alta risoluzione è indispensabile attivare il modo SCREEN 2 o 3.
Per ora ci occuperemo soltanto dei primi due (0 e 1) detti anche modi testo.
Quando accendi il computer ti trovi in SCREEN 0; se hai la pazienza di contare quanti caratteri può contenere una riga scoprirai che sono 37.
Batti ora

```
SCREEN1
```

e ripeti l'operazione di conta...
Adesso ce ne stanno soltanto 29, ma sono più leggibili: la più importante differenza tra i due modi di schermo consiste infatti nella diversa matrice di visualizzazione dei caratteri.
In SCREEN 0 il carattere è iscritto in una griglia di 6x8 punti, mentre in

SCREEN 1 la griglia è di 8x8 e garantisce una maggiore leggibilità.
Introduci:

```
SCREEN 1 : PRINT CHR$ (210)
```

osserva il carattere grafico che appare.
Batti ora:

```
SCREEN 0 : PRINT CHR$ (210)
```

nota la differenza.
L'istruzione SCREEN serve anche a molti altri scopi; uno di questi ha un effetto che lascio scoprire a te.

Prova con:

```
SCREEN ,,0
```

e batti dei tasti a caso....
Il click è sparito; per riattivarlo:

```
SCREEN ,,1
```

WIDTH

L'istruzione WIDTH consente di stabilire il numero di caratteri stampabili su una riga dello schermo.
In SCREEN 0 il numero massimo è 40.
In SCREEN 1 il numero massimo è 32.
Per entrambi i modi di schermo il numero minimo consentito è 1.

```
WIDTH 20
```

fissa in 20 caratteri la riga stampabile sullo schermo.

```
10 SCREEN 0
20 FOR I=40 TO 12 STEP-2
30 WIDTH I
40 PRINT "QUESTA RIGA PROVA L'EFFETTO DI WIDTH"
50 FOR PAUSA=0 TO 333: NEXT PAUSA
60 NEXT I
```

Per tornare alla normalità:

```
WIDTH 37
```

Funzioni di controllo della stampa su video

Nell'ambito delle istruzioni di visualizzazione previste dal BASIC, particolare importanza rivestono tutti i comandi che controllano e modificano, a piacere del programmatore, la posizione del cursore e - di conseguenza - delle scritte sullo schermo.
Abbiamo finora adoperato l'istruzione PRINT in numerose occasioni, utilizzandola per visualizzare tutti i risultati, i messaggi e le scritte che di volta in volta ci sono stati convenienti o necessari.
Ciò che però ancora ci manca è la capacità di controllare completamente questa istruzione, consentendoci per esempio di ottenere in uscita i risultati disposti in una certa posizione dello schermo, oppure ordinati ed incolonnati in un formato non necessariamente impostoci dal nostro calcolatore.
Detto in una parola (peraltro molto usata nel ramo dell'informatica), vogliamo saper formattare le scritte sullo schermo.
Questo è pertanto l'obiettivo delle istruzioni che oggi ci proponiamo di imparare.
La cosa importante da sottolineare è che tutte queste istruzioni non indicano al calcolatore cosa stampare, ma soltanto DOVE stampare.
Capito la differenza?
Vediamole adesso una per una, insieme a qualche esempio esplicativo.

SPC

La funzione SPC () è utilizzata all'interno di istruzioni PRINT per scrivere un certo numero di spazi.
L'argomento numerico assegnato alla funzione specifica il numero di spazi da scrivere.
SPC() permette quindi di spostare su una linea il cursore di quante posizioni si vuole.
L'argomento che fornirai dovrà essere un valore numerico intero compreso tra 0 e 255 (se vi saranno anche cifre decimali, queste verranno eliminate automaticamente); in caso contrario il tuo MSX ti invierà il messaggio di errore:

```
Illegal function
call
```

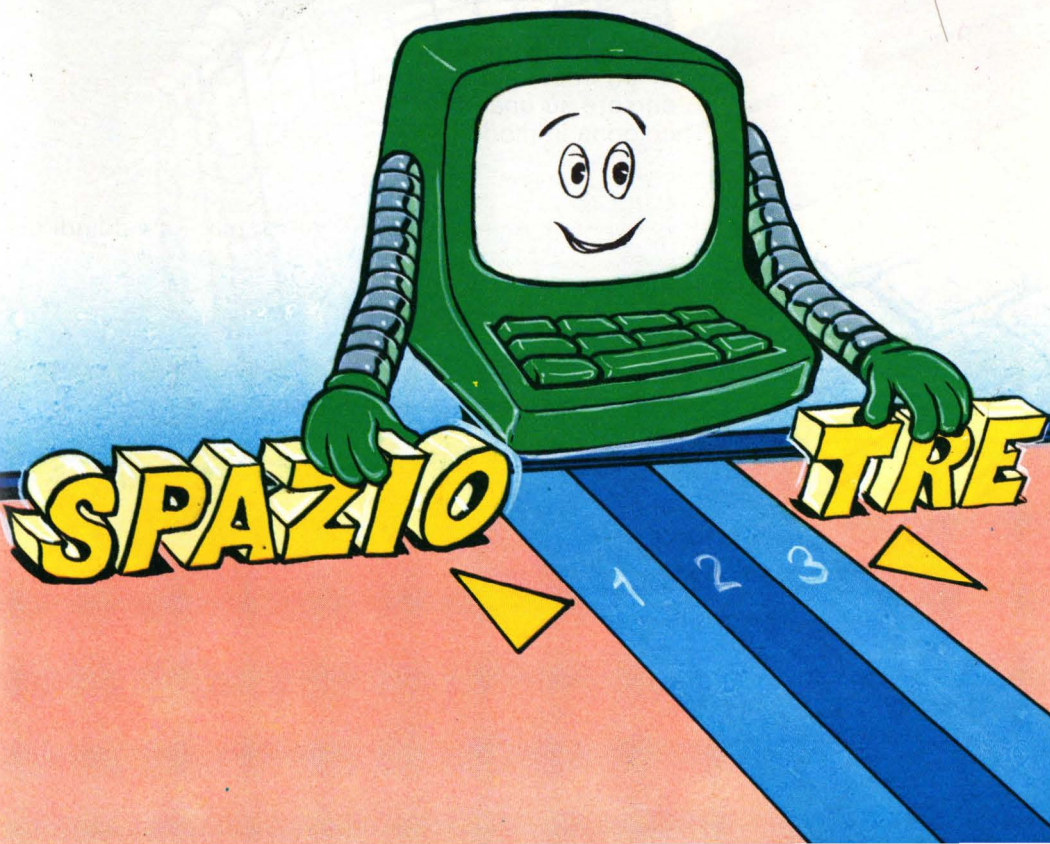
Esempi

```
PRINT "ROSSO"; SPC (4); "DI"; SPC (3); "SERA"
```

Questa istruzione interpone 4 spazi tra le parole ROSSO e DI, mentre tra le parole DI e SERA inserisce 3 spazi.

Nota come dopo gli SPC compaia il punto e virgola. Se infatti vi fosse stata una virgola avremmo avuto anche l'effetto di tale virgola, ottenendo in uscita le parole ancora più distanziate.

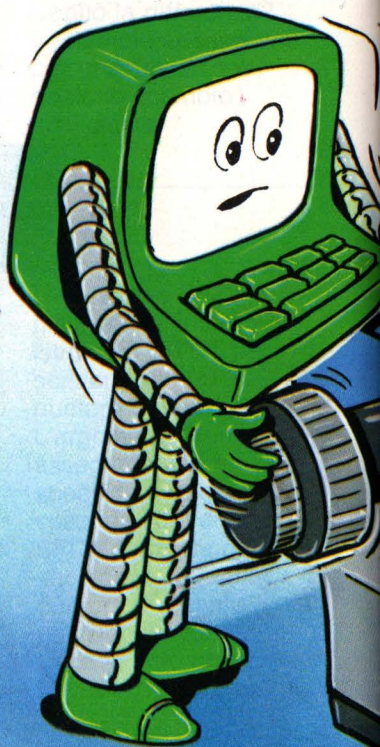
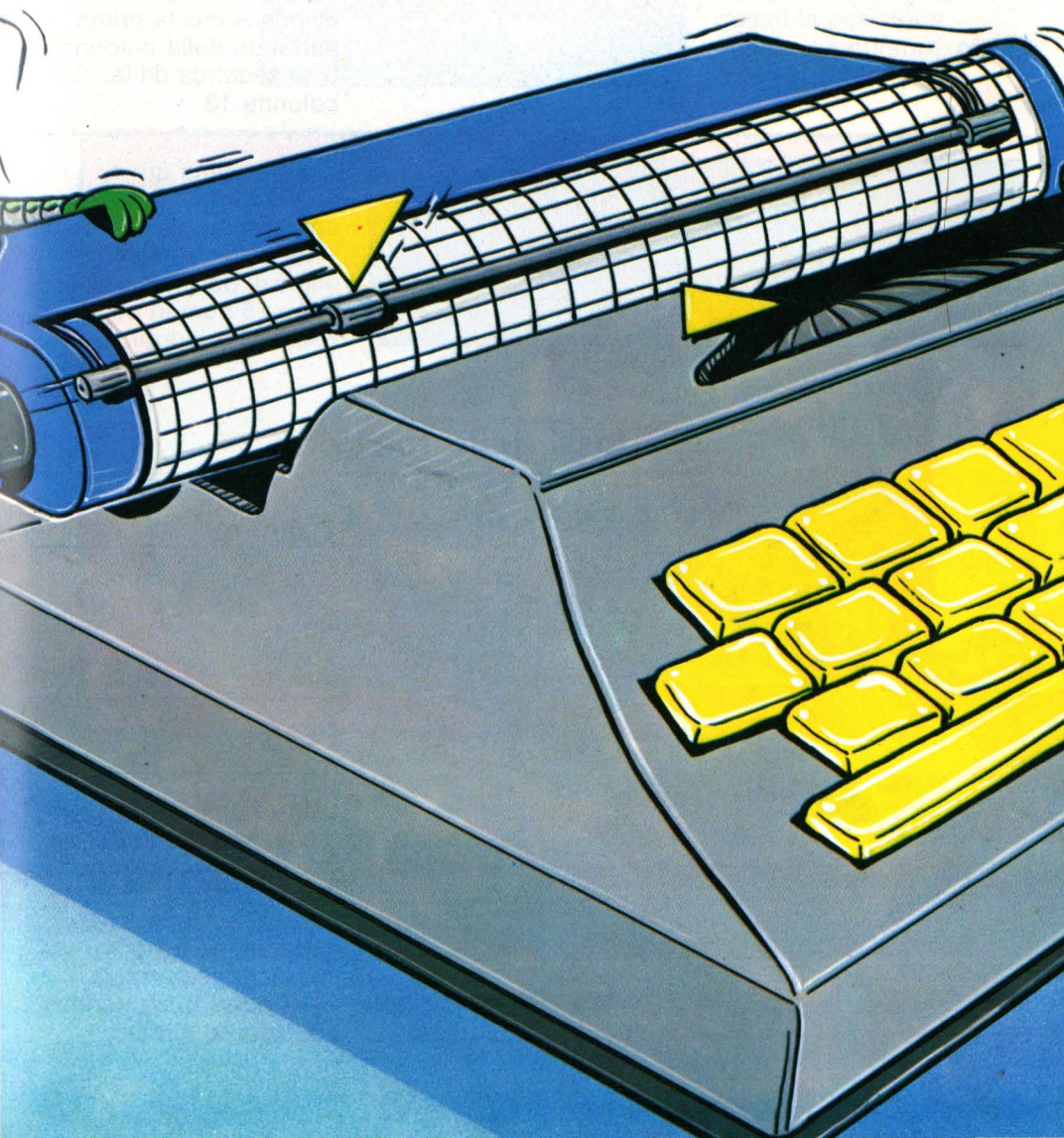
```
10 FOR I=0 TO 18
20 PRINT SPC(I); "CIAO"
30 NEXT
```

Questo breve programma stampa invece una serie di saluti, spostandoli man mano sullo schermo.

Sintassi della funzione

SPC (espressione)

dove espressione deve essere un valore numerico intero compreso tra 0 e 255.

TAB

Anche la funzione TAB() si usa solo all'interno delle PRINT: essa opera infatti come la tabulazione di una normale macchina da scrivere.
TAB() si distingue da SPC() in quanto sposta il cursore su una certa colonna in modo assoluto (cominciando cioè a contare dalla colonna numero 0), mentre SPC() lo sposta in un certo numero di colonne in modo relativo (contando cioè gli spazi

partendo dall'attuale posizione di stampa). La funzione TAB() viene normalmente utilizzata per allineare le stampe in colonne verticali, incolonnandole in punti prestabiliti.
Supponi infatti di dover visualizzare alcuni dati sullo schermo del tuo video, allineati in un certo ordine. Grazie a TAB() puoi evitare i noiosi (e talvolta complicati) calcoli per incolonnare esattamente tutti gli elementi.

Esempi

```
PRINT NOME$; TAB(13); COGNOME$
```

Con questa istruzione, indipendentemente dalla lunghezza della stringa contenuta in NOME$, si avrà la stampa delle due variabili con questa disposizione: la prima partendo dalla colonna 0, la seconda dalla colonna 13.

```
10 PRINT TAB(2); "NUMERO"; TAB(12);
   "QUADRATO"
20 FOR I=1 TO 15
30 PRINT TAB(4);I;TAB(15);I↑2
40 NEXT
```

Il programma qui a fianco mostra una semplice applicazione di TAB: stampa infatti, allineati per colonna, i primi 15 numeri con i rispettivi quadrati.

Sintassi della funzione

TAB (espressione)

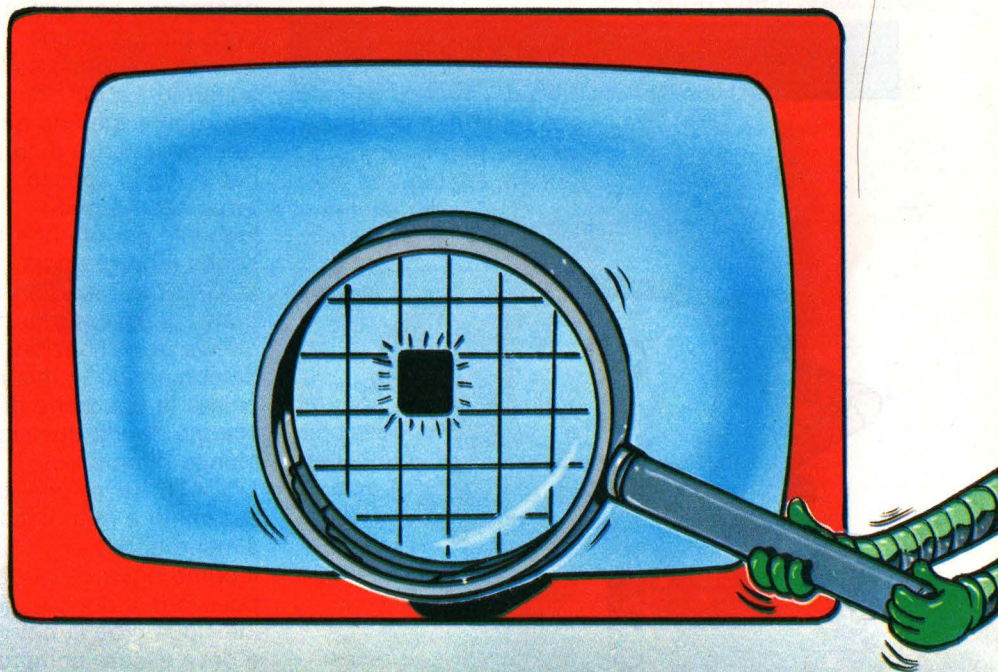
dove espressione è un valore numerico compreso tra 0 e 255.

POS

La funzione POS() restituisce la posizione attuale del cursore in senso orizzontale, cioè la colonna sulla quale verrà stampato il carattere successivo. Restituisce quindi come risultato un numero

compreso tra 0 e 39. L'argomento della funzione (strano, ma vero!) non ha alcuna importanza; deve però essere sintatticamente corretto (normalmente si utilizza lo zero).
Ad esempio, il seguente breve programma,

```
10 PRINT SPC(5);"CANE";SPC(5)
20 PRINT POS(0)
```

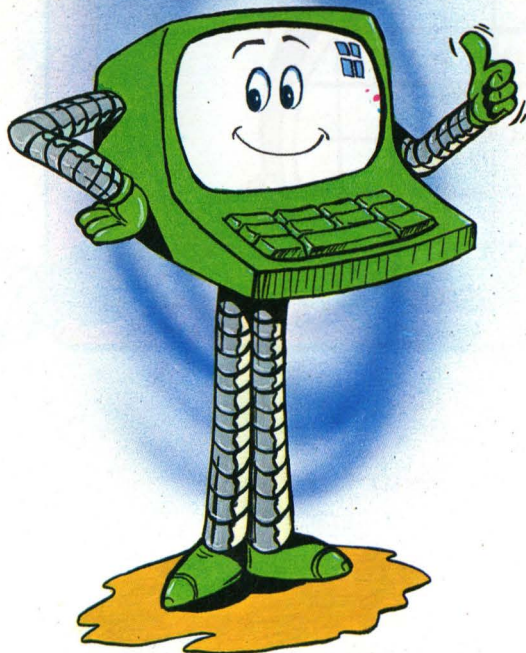
provocherà la visualizzazione della parola CANE preceduta e seguita da 5 spazi (che naturalmente risultano invisibili all'occhio umano).
La riga 20 stamperà quindi la posizione attuale del cursore, cioè 5+4+5 = 14.

Sintassi della funzione

POS (argomento)

dove argomento è un valore numerico puramente formale, senza cioè alcuna specifica funzione.

CSRLIN

Del tutto analogo a POS, questa funzione se ne differenzia dal fatto che rileva la posizione del cursore in senso verticale.
Di fatto restituisce un numero compreso tra 0 e 23 che individua la linea su cui è posizionato il cursore.
Non richiede argomento.

```
CLS : PRINT : PRINT : PRINT : PRINT  CSRLIN
```

restituisce come risultato 3 ad indicare che il cursore all'istante dell'esecuzione, era posizionato sulla quarta riga dall'alto.

LOCATE

In moltissimi casi è necessario scrivere una certa stringa in una precisa posizione sullo schermo. Le funzioni e istruzioni viste fino ad ora consentono di farlo, ma in modo spesso arraginoso e poco leggibile per chi non ha scritto il programma.
LOCATE è l'istruzione in grado di eliminare tutte queste complicazioni determinando in modo chiaro la posizione di stampa di una qualsiasi stringa di caratteri.
LOCATE ha tre possibili parametri e ne richiede almeno uno.
Il primo riguarda la posizione x del cursore; individua cioè la colonna e deve essere compreso tra 0 e 39.
Il secondo indica la riga fornendo la coordinata y del cursore.

Il terzo è in realtà un interruttore in grado di accendere (1) o spegnere (0) il cursore.

```
LOCATE 12, 6, 1 : FOR I=1TO2000 :NEXT
```

posiziona il cursore alla tredicesima colonna (N. 12) della settima riga (N. 6) visualizzandolo.

```
LOCATE, 12, 1 : FOR I=1 TO 2000 : NEXT
```

posiziona il cursore alla colonna corrente (cioè in quella in cui già si trova) ma nella riga N. 12 e lo visualizza.

```
LOCATE 10, 20, 1 : PRINT"PROVA";:FORI=1
TO 2000 : NEXT
```

stampa a partire dalla undicesima colonna della ventunesima riga la stringa PROVA che terminerà con il cursore in evidenza.

Sintassi dell'istruzione

LOCATE colonna, riga, cursore.
I parametri devono essere valori numerici sotto forma di costante o variabile.

RND

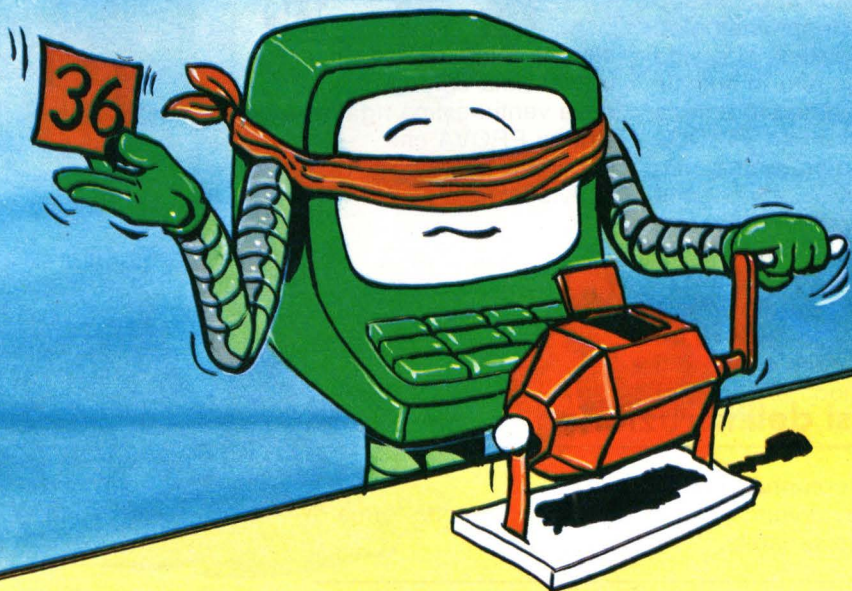
Non è raro trovare casi in cui occorra, nell'ambito di un programma, disporre di numeri casuali.
Potresti, ad esempio, aver bisogno di un algoritmo che simuli l'estrazione a sorte di un numero, o il lancio dei dadi o ancora l'uscita delle carte da gioco dal mazzo.

LINGUAGGIO

La funzione RND ti risolve egregiamente questo tipo di problemi: crea, infatti, dei numeri casuali (random in inglese) compresi tra 0 e 1 (0 < = R < 1).
Il tuo MSX produce una sequenza di numeri casuali eseguendo dei calcoli su di un numero di partenza chiamato seme, generato al momento dall'accensione.
Poiche i numeri generati da RND sono il risultato di una complessa serie di calcoli, è più corretto parlare di numeri pseudocasuali.
L'argomento di RND controlla il seme della funzione e determina il punto di partenza della serie dei numeri casuali.
Il formato della funzione è:

```
RND (S)
```

dove S rappresenta un qualsiasi numero reale. Il valore di S determina il comportamento della funzione:

1) Se S è un numero positivo RND restituisce il numero casuale successivo nella sequenza.
2) Con S=0 si ottiene e viene ripetuto l'ultimo numero nella sequenza.
3) Con S minore di 0 viene generata una nuova serie di numeri correlati all'argomento fornito.

Ad esempio, il programma che segue produrrà sempre la stessa sequenza di numeri indipendentemente da quando e quante volte darai il RUN:

```
10 FOR I=1TO5
20 PRINT RND (0)
30 NEXT I
```

Per ottenere dunque serie diverse di numeri è necessario ricorrere ad un argomento minore di 0 che vari in continuazione.
Una ottima opportunità di un valore in continuo mutamento è offerta dalla funzione TIME.
Essa restituisce il valore dell'orologio del computer il quale parte da 0 al mometo dell'accensione e viene incrementato ogni cinquantesimo di secondo.
Rendendolo minore di zero per mezzo del segno — (meno) ecco pronto il seme per una perfetta sequenza pseudocasuale di numeri.

```
10 FOR I=1TO 5
20 PRINT RND (-TIME)
30 NEXT I
```

Ad ogni RUN vedrai apparire numeri sempre diversi.
Poiché la gamma dei numeri compresi tra 0 ed 1 non è adeguata alla maggior parte delle applicazioni, usa la seguente formula per stabilire tu stesso il "range" entro il quale dovranno essere prodotti i numeri casuali:

```
LET R = INT ((LS - LI + 1) * RND (1)) + LI
```

Dove R sta per numero Random, LS è il Limite Superiore del "range" ed LI il Limite Inferiore.
Per numeri Random interi compresi tra 1 e LS (variabile numerica da definire precedentemente), applicando la formula avrai:

```
LET R = INT (LS * RND (1)) + 1
```

Sintassi della Funzione

RND (Espressione Numerica)

L'arte di stampare

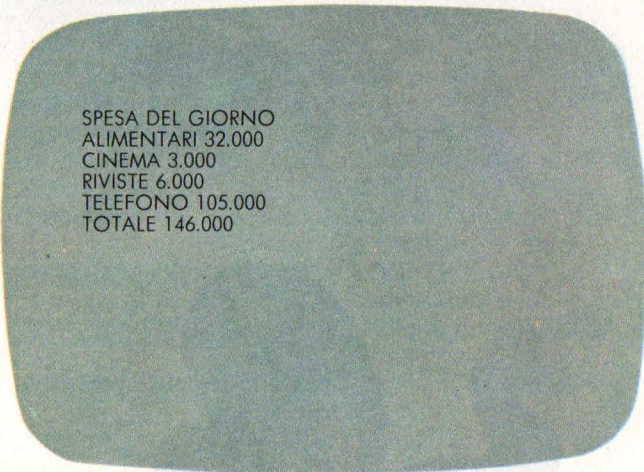
Una delle cose più importanti di un buon programma è quella di presentare i dati da visualizzare in modo preciso, chiaro e piacevole.
Informazioni sostanzialmente corrette e ineccepibili, possono vedere sminuito il loro pregio a causa di una presentazione disordinata o poco leggibile.
Ci sono dei piccoli accorgimenti, attenendoti ai quali puoi ottenere facilmente dei buoni risultati.
1) Usa molto spesso l'istruzione CLS: evita cioè di ammassare grandi quantità di dati su di una sola schermata.
Per evitare di sottrarre informazioni interessanti all'utente chiedi il permesso di voltare pagina prima di cancellare.

```
10 SCREEN1 : WIDTH 32 : KEYOFF
20 LOCATE 2,6 : PRINT"PRIMA DI PASSARE
   ALLE PROSSIME"
30 PRINT" INFORMAZIONI..."
40 LOCATE 9,24 : PRINT "PREMI UN TASTO";
50 A$=INKEY$ : IFA$= " "THEN50
60 CLS : REM
```

Evidenzia le informazioni importanti lasciando almeno una riga vuota prima e dopo di esse.

```
10 CLS: PRINT"N N N N N N N N N N N N N N N N N N
   N N N N N N N N N N"
20 PRINT"N N N N N N N N N N N N N N N N N N
   N N N N N N N N N N"
30 PRINT"U U U U U U U U U U U U U U U U U U
   U U U U U U U U U U"
40 PRINT
   50 PRINT "QUESTA RISULTA EVIDENZIATA"
60 PRINT
80 PRINT"S S S S S S S S S S S S S S S S S S
   S S S S S S S S S S"
90 PRINT"S S S S S S S S S S S S S S S S S S
   S S S S S S S S S S"
```

3) Presta attenzione alla marginatura sinistra. Seleziona il formato di schermo più adatto alla rappresentazione dei dati.
4) Scegli dei colori di inchiostro, carta e bordo tali da ottenere la massima nitidezza di scrittura.
5) Rispetta gli incolonnamenti di numeri e testi specie nelle tabelle.
Tutto questo, dopo aver fatto un po' di pratica con funzioni come TAB e istruzioni come LOCATE e COLOR ti sarà chiaro e molto più facile di quanto possa immaginare.

L'errore è sempre in attesa: malignamente aspetta una tua distrazione, anche minima. Non dargli mai la possibilità di sogghignare. Impara la sintassi delle istruzioni. Realizza programmi in forma modulare, facilmente leggibili, e molto documentati. Nella digitazione, poi, metti la massima attenzione: anche una virgola al posto sbagliato, compromette la buona riuscita del programma.

Lista della spesa

Il programma che segue somma le voci di una spesa giornaliera. Per terminare l'input dei dati, introduci 0, 0. Nota come, per ottenere una corretta visualizzazione dei dati, si sia fatto ricorso alla funzione TAB e all'istruzione LOCATE con parametri variabili.

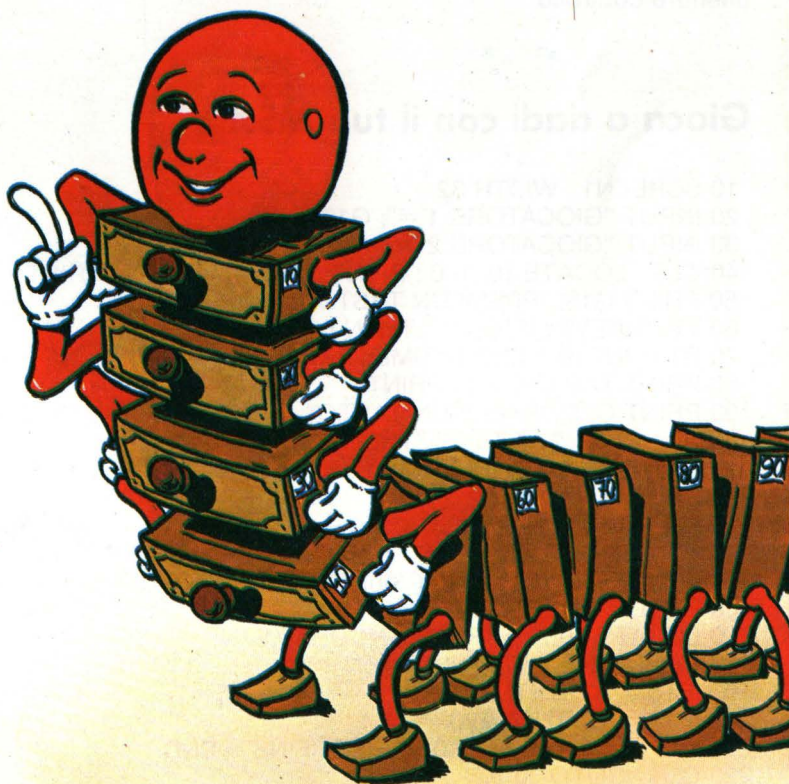
Da rilevare inoltre l'artificio che consente di riproporre l'INPUT sulla stessa linea.
Per cancellare i vecchi dati basta stampare su di essi una stringa di spazi memorizzata nella variabile S$.

```
SPESA DEL GIORNO
ALIMENTARI 32.000
CINEMA 3.000
RIVISTE 6.000
TELEFONO 105.000
TOTALE 146.000
```

```
SPESA DEL GIORNO

ALIMENTARI..........................32.000
CINEMA...............................3.000
RIVISTE..............................6.000
TELEFONO...........................105.000

TOTALE                             146.000
```

```
10 SCREEN1 : WIDTH32 : COLOR 15,1,6
20 T=0 : R=4 : C=2 : Z=T : PRINT TAB(7); "LISTA DELLA SPESA"
30 S$ = "                              ".: REM 31 SPAZI
40 LOCATE Z,C : PRINT S$ : LOCATE Z, C : INPUT "DESCR, IMPORTO"; D$, I
50 IFI = 0 THEN LOCATE C, R+2 : PRINT "TOTALE"; TAB(20) ; T ; END
60 T = T + I : R = R + 1 : LOCATE C, R : PRINT D$ ; TAB (20) ; I
70 GOTO 40
```

VIDEOESERCIZI

Usando la funzione TAB, ricerca un nuovo metodo per stampare la tabellina di un numero, in modo che le unità, le decine, ecc. dei vari prodotti risultino perfettamente incolonnate. Ecco, per aiutarvi, un esempio con quella del 7.

```
10 C=9 : S=C : REM LE UNITA' VANNO STAMPATE ALLA COLONNA 10
20 CLS
30 FOR V=1 TO 10
40 P=V * 7
50 IFP > 9 THEN S=C-1
60 PRINT TAB (S); P
70 S=C
80 NEXT
```

Prova a sostituire la linea 30 con : 30 FOR V = 1 TO 20.
Suggerimento: per incolonnare le centinaia, introduci tra la linea 50 e la 60 un ulteriore controllo.

Gioca a dadi con il tuo MSX

```
 10 SCREEN1 : WIDTH 32
 20 INPUT "GIOCATORE 1 ="; G1$
 30 INPUT "GIOCATORE 2 ="; G2$
 40 CLS : LOCATE 10, 1, 0 : PRINT "DADOMATTO" : PRINT
 50 PRINT G1$; "PREMI UN TASTO";
 60 T$=INKEY$ : IFT$= " " THEN 60
 70 R1 = INT (6 * RND (- TIME)) + 1
 80 PRINT TAB (28); R1 : PRINT
 90 PRINT G2$; "PREMI UN TASTO";
100 T$= INKEY$ : IFT$= " "THEN 100
110 R2 = INT (6 * RND (-TIME)) +1
120 PRINT TAB (28); R2
130 LOCATE 10,15,1
140 IFR1 = R2 THEN PRINT "PARTITA PARI" : GOTO 170
150 IFR1 > R2 THEN PRINT "VINCE"; G1$ : GOTO 170
160 PRINT "VINCE" : G2$
170 LOCATE 10,18 : PRINT "ANCORA? S/N";
180 T$=INKEY$ : IFT$= " " THEN 180
190 IFT$= "s" THEN 220
200 IFT$= "S" THEN 220
210 CLS : LOCATE 14,12,0 : PRINT "FINE" : END
220 CLS : GOTO 40
```